



Soft Aspects

RIAs & Beyond



New light of Rich Internet Applications

WebGalileo Faces© TreeTable Component

Quick Start

Table of Contents

1 Overview.....	3
2 Setup.....	3
3 Assembling the component.....	7

1 Overview

The WebGalileo Faces TreeTable component extends Tree component and can display one of the columns as Tree. ColumnEditor class for that column should be **com.softaspects.framework.galileo.renderers.html.treetable.TreeEditor**.

TreeTable component supports sorting and table data editing . The component sorts data belonging to a given tree node firstly . Then it sorts all top level branch nodes.

2 Setup

To assemble the TreeTable the following steps should be accomplished :

1. Add wgf-core.jar, wgf-faces.jar and wgf-tags.jar libraries from WebGalileo Faces deliverable to WEB-INF/lib directory of a target web application.
2. Add webgalileo.key and components-faces-config.xml from WebGalileo Faces deliverable to WEB-INF directory of a target web application.
3. Develop TreeTable model beans. For Tree column editorModel should be filled by corresponding Tree dataModel: every treeItem should be added to editorModel.

```
TreeTableTreeDataModelBean treeDataModelBean = new
TreeTableTreeDataModelBean();
fillData(treeDataModelBean);
```

FillData function may look like :

```
private void fillData(BaseTreeDataModel item) {
    if (item instanceof TreeItem) {
        addValue(item);
        if (!(((TreeItem) item).getParentEntity() instanceof TreeItem))
            ((TreeItem) item).setOpened(true);
    }
    for (int i = 0; i < item.getSize(); i++)
        if (item.getElementAt(i).getChildrenEntites() != null)
            fillData((BaseTreeDataModel) item.getElementAt(i));
        else
            addValue(item.getElementAt(i));
}
```

4. Define the managed beans:

```

<managed-bean>
  <managed-bean-name>treeTableDataModel</managed-bean-name>
  <managed-bean-
class>testingapplication.treetable.TreeTableDataModelBean</managed-bean-
class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>treeTableTreeDataModel</managed-bean-name>
  <managed-bean-
class>testingapplication.treetable.TreeTableTreeDataModelBean</managed-bean-
class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>treeTableColumnModel</managed-bean-name>
  <managed-bean-
class>testingapplication.treetable.TreeTableColumnModelBean</managed-bean-
class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>treeTableHeader</managed-bean-name>
  <managed-bean-
class>testingapplication.treetable.TreeTableHeaderBean</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>treeTableInterfaceManager</managed-bean-name>
  <managed-bean-
class>testingapplication.treetable.TreeTableInterfaceManagerBean</managed-
bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>myTreeInterfaceManager</managed-bean-name>
  <managed-bean-
class>testingapplication.tree.TreeInterfaceManagerBean</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>

```

5. Reference the models defined in the previous step in the corresponding tag when assembling the component

```
<treetable:treetable id="treetable1"
    columnSelectionType="multiple"
    rowsPerPageValues="5,10,20,50"
    serverSideAction="true"
    cellPadding="0"
    cellSpacing="0" >
  <treetable:dataModel>
    <table:dataModel beanName="#{treeTableDataModel}" scope="session"/>
    <tree:dataModel beanName="#{treeTableTreeDataModel}" scope="session"/>
  </treetable:dataModel>
  <table:columnModel beanName="#{treeTableColumnModel}" scope="session"/>
  <table:header beanName="#{treeTableHeader}"/>
  <table:interfaceManager beanName="#{treeTableInterfaceManager}"/>
  <tree:interfaceManager beanName="#{myTreeInterfaceManager}"/>
</treetable:treetable>
```

6. **Create Table data model.** The row values for tree column can be any but **NOT NULL** , for example it can be any String value .

7. **Create Tree data model.**

8. **Create TreeTable data model:**

```
<treetable:dataModel>
  <table:dataModel beanName="#{treeTableDataModel}" scope="session"/>
  <tree:dataModel beanName="#{treeTableTreeDataModel}" scope="session"/>
</treetable:dataModel>
```

9. **Create Table interface manager and set editor data:**

```
<table:interfaceManager beanName="TreeTableInterfaceManager"
    scope="session"
    overwriteExistent="true"
    ...
    columnIndex="0"/>
</table:interfaceManager>
```

10. Create Table column model and set column editor class for tree column:

```

<table:columnModel beanName="treeTableColumnModel"
    columnSelectionAllowed="false"
    scope="session"
    overwriteExistent="false">
    <table:column ID="Tree" modelIndex="0" width="90pt" sortSupported="false"
tableValueChangedAction="false">
        <table:setColumnEditor
editorClass="com.softaspects.framework.galileo.renderers.html.treetable.TreeE
ditor"/>
    </table:column>
    ...
</table:columnModel>

```

11. Create Tree InterfaceManager to setup tree interface :

```

<tree:interfaceManager beanName="TreeInterfaceManager"
    fontFamily="Comic Sans MS"
    fontStyle="oblique"
    fontSize="24"
    ...
</tree:interfaceManager>

```

12. Create TreeTable component and set datamodel, column model and interface managers:

```

<treetable:treetable id="treetable1"
    columnSelectionType="multiple"
    rowsPerPageValues="5,10,20,50"
    cellPadding="0"
    cellSpacing="0" >
    <treetable:dataModel>
        <table:dataModel beanName="#{treeTableDataModel}" scope="session"/>
        <tree:dataModel beanName="#{treeTableTreeDataModel}" scope="session"/>
    </treetable:dataModel>
    <table:columnModel beanName="treeTableColumnModel" scope="session"/>
    <table:header beanName="treeTableHeader"/>
    <table:interfaceManager beanName="TreeTableInterfaceManager"/>
    <tree:interfaceManager beanName="TreeInterfaceManager"/>
</treetable:treetable>

```

3 Assembling the component

1. Tag definition:







```
<treetable:treetable ajaxRefreshEnabled="true" id="treetable1"
columnSelectionType="multiple"
rowSelectionType="single"
rowsPerPageValues="5,10,20,50"
serverSideAction="true"
tableRowsPerPageChangeListener="#{myTableRowsPerPageChangeListener}"
tableValueChangedListener="#{myTableValueChangedListener}"
tableNavigationSelectedListener="#{myTableNavigationSelectedListener}"
tableSortSelectedListener="#{myTableSortSelectedListener}"
cellPadding="0" cellSpacing="0">

<table:columnModel beanName="#{treeTableColumnModel}" scope="session"/>
<treetable:dataModel>

<table:dataModel beanName="#{treeTableDataModel}" scope="session"/>
<tree:dataModel beanName="#{treeTableTreeDataModel}" scope="session"/>
</treetable:dataModel>

<table:header beanName="#{treeTableHeader}"/>
<table:interfaceManager beanName="#{treeTableInterfaceManager}"/>
</treetable:treetable>
```

2. Produced output :

Tree ▾	Column1 ▾	Column3	Column4	Column5	Column7
 User	<input type="text" value="0"/>	-20%	\$-816.64	Sep 30, 2001	0.525
 Video	<input type="text" value="1"/>	40%	\$218.52	Apr 21, 2003	0.718
 Empty Video Folder	<input type="text" value="5"/>	84%	\$479.13	Dec 1, 1998	0.313
 Storages	<input type="text" value="6"/>	-46%	\$-864.07	Aug 22, 1997	0.113
 Video and Photography	<input type="text" value="13"/>	75%	\$794.73	Mar 30, 2003	0.269
 Sound	<input type="text" value="16"/>	-48%	\$-800.41	Dec 5, 2005	0.653

Rows Per Page : Total : 6